2 0 0 9년 8월

석사학위 논문

Adaptive Multi-dimension Segmentation of HSV Cone for Image Retrieval

조선대학교 대학원

정보통신공학과

Muhammad Riaz

Waqas

2009년 8월 석사학위논문

Adaptive Multi-dimension Segmentation of HSV Cone For Image Retrieval

Muhammad Riaz

Adaptive Multi-dimension Segmentation of HSV Cone For Image Retrieval

2 0 09 년 8월 25일

조선대학교 대학원

정보통신공학과

Muhammad Riaz

Adaptive Multi-dimension Segmentation Of HSV Cone for Image Retrieval

지도교수 박종안

이 논문을 이학 석사학위신청 논문으로 제출함

2 0 0 9 년 8월

조선대학교 대학원

정보통신공학과

Muhammad Riaz

Muhammad Riaz의 석사학위논문을 인준함

위원장	한승조 (인)
위 원	변재영 (인)
위 원	박종안 (인)

2 0 09년 8월

조선대학교 대학원

Table of Contents (목차)

List of Figures

List of Tables

ABSTRACT

Adaptive Multi-dimension Segmentation of HSV Cone for Image Retrieval

Muhammad, Riaz
Advisor : Prof. Park, Jong-An, Ph. D
College of Electronics and Information Engineering
Graduate School of Chosun University

The study mainly covers the subject of content based image retrieval (CBIR) using adaptive segmentation of HSV cone. Image segmentation is a low-level task that consists on partitioning the image into homogeneous regions, according to some criteria. It is a crucial step in computer vision systems involving image processing (e.g. content-based image retrieval) where the challenge is to perform an image segmentation with some semantic meaning. Although promising results are presented in many papers, genericity is still not proven. In fact, many of these approaches suffer from a subjective tuning of key parameters. This problem also occurs in many vision systems where segmentation stage is narrowly tuned according to the application domain specities by a human expert in image processing.

Some specialized applications for image based query and retrieval are researched and some useful results are produced. Many references have been studied in order to rank various notable features of images, which prove key indices or identification marks of various images. Under the light of most reputable and most remarkable research, the features were rigorously tested and confirmed for their solidity. Simulation was performed using Java and MATLAB.

I. Introduction

A. Overview

In the start human visual system is investigated, then how a computer can be made to sense pictorial data and then how can the images be processed.

B. Human and computer vision

A computer vision system processes images acquired from an electronic camera, which is like the human vision system where the brain processes images derived from the eyes. Computer vision is a rich and rewarding topic for study and research for electronic engineers, computer scientists and many others. Increasingly, it has a commercial future. There are now many vision systems in routine industrial use: cameras inspect mechanical parts to check size, food is inspected for quality, and images used in astronomy benefit from computer vision techniques. Forensic studies and biometrics (ways to recognize people) using computer vision include automatic face recognition and recognizing people by the 'texture' of their irises. These studies are paralleled by biologists and psychologists who continue to study how our human vision system works.

A selection of digital images is given in Figure 1.1, these images comprise a set of points or picture elements stored as an array of numbers in a computer. To recognize faces, based on an image such as Figure 1.1(a), we need to be able to analyze constituent shapes, such as the shape of the nose, the eyes, and the eyebrows, to make some measurements to describe, and then recognize, a face. Figure 1.1(b) is an ultrasound image of the carotid artery which is near the side of the neck and supplies blood to the brain and the face, taken as a cross-section through it. The top region of the image is near the skin; the bottom is inside the neck. The image arises from combinations of the reflections of the ultrasound radiation by tissue. This image comes from a study aimed to produce three-dimensional models of arteries, to aid vascular surgery. Note that the image is very noisy, and this obscures the shape of the elliptical artery. Remotely sensed images are often analyzed by their texture content. The perceived texture is different between the road junction and the different types of foliage seen in Figure 1.1(c). Finally, Figure 1.1(d) is a Magnetic Resonance Image (MRI) of a cross-section near the middle of a human body. The chest is at the top of the image, and the lungs and blood vessels are the dark areas, the internal organs and the fat appear grey. MRI images are in routine medical use nowadays, owing to their ability to provide high quality images.

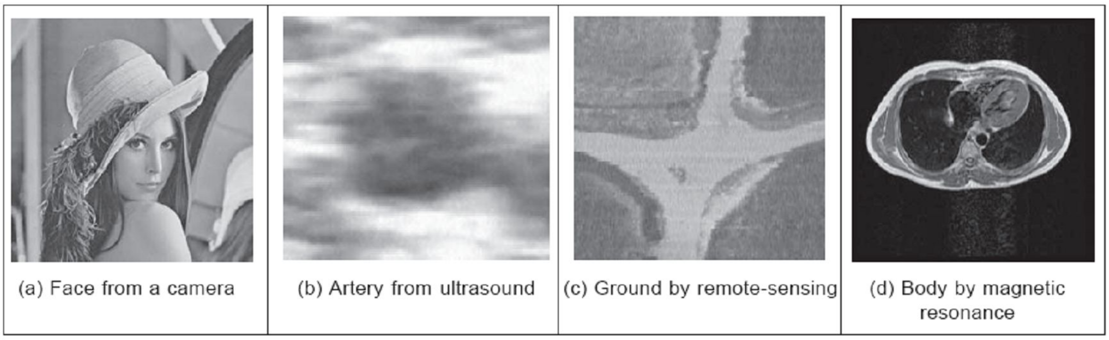

Figure 1.1: Real images from different sources

Synthesized images are good for evaluating techniques and finding out how they work, and some of the bounds on performance. Two synthetic images are shown in Figure 1.2. Figure 1.2(a) is an image of circles that were specified mathematically. The image is an ideal case: the circles are perfectly defined and the brightness levels have been specified to be constant. This type of synthetic image is good for evaluating techniques which find the borders of the shape , the shape itself and even for making a description of the shape. Figure 1.2(b) is a synthetic image made up of sections of real image data. The borders between the regions of image data are exact, again specified by a program. The image data comes from a well-known texture database, the Brodatz album of textures. This was scanned and stored as computer images. This image can be used to analyze how well computer vision algorithms can identify regions of differing texture.

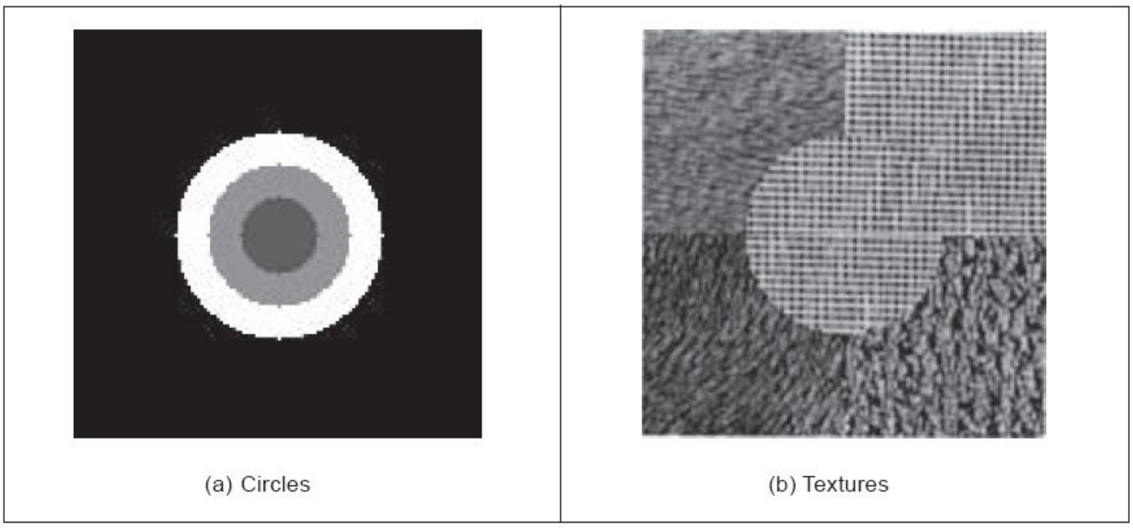

Figure 1.2: Examples of synthesised images

C. Computer Vision in Detail

Computer vision is the science and technology of machines that see. As a scientific discipline, computer vision is concerned with the theory for building artificial systems that obtain information from images. The image data can take many forms, such as a video sequence, views from multiple cameras, or multi-dimensional data from a medical scanner.

As a technological discipline, computer vision seeks to apply the theories and models of computer vision to the construction of computer vision systems. Examples of applications of computer vision systems include systems for:

- Controlling processes e.g. an industrial robot or an autonomous vehicle.
- Detecting events e.g. for visual surveillance or people counting.
- Organizing information e.g. for indexing databases of images and image sequences.
- Modeling objects or environments e.g. industrial inspection, medical

image analysis or topographical modeling.

- Interaction e.g. as the input to a device for computer-human interaction.

Sub-domains of computer vision include scene reconstruction, event detection, tracking, object recognition, learning, indexing, motion estimation, and image restoration. Figure 1.3 shows the related fields of computer vision.

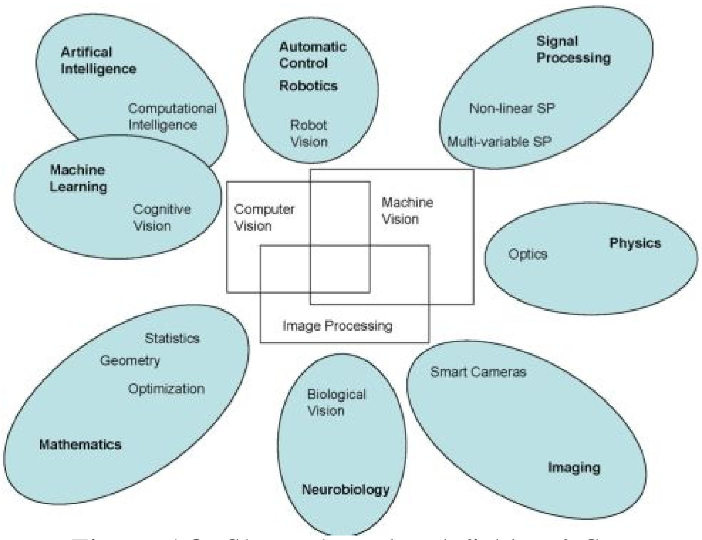

Figure 1.3: Show the related fields of Computer Vision

1. Typical Tasks of Computer Vision

Some examples of typical computer vision tasks are presented below.

(1) Recognition

The classical problem in computer vision, image processing and machine vision is that of determining whether or not the image data contains some specific object, feature, or activity. This task can normally be solved robustly and without effort by a human, but is still not satisfactorily solved in computer vision for the general case: arbitrary objects in arbitrary situations. The existing methods for dealing with this problem can at best solve it only for specific objects, such as simple geometric objects e.g. polyhedrons, human faces, printed or hand-written characters, or vehicles, and in specific situations, typically described in terms of well-defined illumination, background, and pose of the object relative to the camera.

(2) Motion

Several tasks relate to motion estimation, in which an image sequence is processed to produce an estimate of the velocity either at each points in the image or in the 3D scene. Examples of such tasks are:

- Egomotion: determining the 3D rigid motion of the camera.
- Tracking: following the movements of objects (e.g. vehicles or humans).

(3) Scene Reconstruction

Given one or (typically) more images of a scene, or a video, scene reconstruction aims at computing a 3D model of the scene. In the simplest case the model can be a set of 3D points. More sophisticated methods produce a complete 3D surface model.

(4) Image Restoration

The aim of image restoration is the removal of noise, sensor noise, motion blur, etc., from images. The simplest possible approach for noise removal is various types of filters such as low-pass filters or median filters. More sophisticated methods assume a model of how the local image structures look like, a model which distinguishes them from the noise. By first analysing the image data in terms of the local image structures, such as lines or edges, and then controlling the filtering based on local information from the analysis step, a better level of noise removal is usually obtained compared to the simpler approaches.

The focus of this study is also on Content Based Image Retrieval, which is briefly explained below.

D. Content Based Image Retrieval

In many areas of commerce, government, academia, and hospitals, large collections of digital images are being created. Many of these collections are the product of digitizing existing collections of analogue photographs, diagrams, drawings, paintings, and prints. Usually, the only way of searching these collections was by keyword indexing, or simply by browsing. Digital images databases however, open the way to content-based searching. A number of other overviews on image database systems, image retrieval, or multimedia information systems have been published. A number of keyword-based general WWW search engines allow to indicate that the media type must be images, see for example http://hotbot.lycos.com/, and http://www.nbci.com/. A number of other general search engines are more specifically for images, such as http://isurf.yahoo.com/ or the multimedia searcher of http://multimedia.lycos.com/, but they are still only keyword based. For example, http://sunsite.berkeley.edu/ImageFinder/ provides a list of such collections as a tool to help teachers locate historical photographs from collections around the world. AltaVistaAltaVista Photofinder is a search engine that allows content-based image retrieval, both from special collections, and from the Web. There is something about Munch's "The Scream" or Constable's "Wivenoe Park" that no words can convey. It has to be seen. The same holds for a picture of the Kalahari Desert, a dividing cell, or the facial expression of an actor playing King Lear. It is beyond words. Try to imagine an editor taking in pictures without seeing them or a radiologist deciding on a verbal description. Pictures have to be seen and searched as pictures: by objects, by style, by purpose.

At the current stage of content-based image retrieval research, it is interesting to look back toward the beginning and see which of the original ideas have blossomed, which haven't, and which were made obsolete by the changing landscape of computing. In February 1992, the US National Science Foundation (USNSF) organized a workshop in Redwood, California, to identify major research areas that should be addressed by researchers for visual information management systems that would be useful in scientific, industrial, medical, environmental, educational, entertainment, and other applications [31]. In hindsight, the workshop did an excellent job of identifying unsolved problems that researchers should have undertaken. In particular, the workshop correctly stated that "Visual Information Management Systems should not be considered as an application of the existing state of the art, in computer vision and databases, to manage and process images and that computer vision researchers should identify features required for interactive image understanding, rather than their discipline's current emphasis on automatic techniques" (emphasis added). As possible application fields, the workshop considered mainly Grand Challenge problems, such as weather forecasting, biological modeling, medical images, satellite images, and so on. Undoubtedly, the participants saw enough to justify the use of the large computational and storage capacity necessary for visual databases. This in 1992. The workshop was preceded by many years by the Conference on Database Applications of Pictorial Applications, held in Florence in 1979, probably one of the first conferences of that kind [1]. In the introduction, it was said that: This has facilitated the advancement of integrated databases on the one hand, of and graphical and image processing (in brief: pictorial) applications on the other. Then, the author proceeds to complain that: Developments in these two fields have traditionally been unrelated, an observation still very much valid today.

Just after the USNSF workshop, the Mosaic Internet browser was released, spawning the Web revolution that very quickly changed all cards. In the same era, a host of new digital vision sensors became available. The number of images that the average user could reach increased dramatically in just a few years. Instantly, indexing tools for the Web or digital archives became urgent. For a system-based review [2].

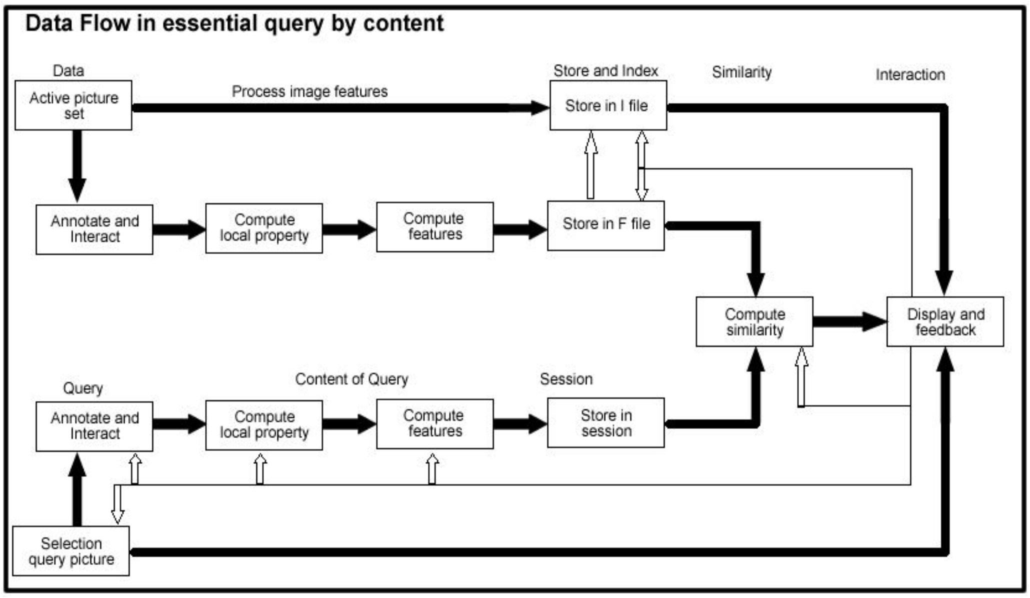

Figure 1.4: Basic algorithmic components of query by pictorial example

Target-, category- and association-search in Image retrieval

	Target	Category	Association
Object Goal	1 specific object	an arbitrary object from 1 specific class	not defined at start
Query by example	1 ... N objects	1 ... N objects with class labels	N objects plus association
Similarity	feature based	class driven	session specific
Events in F-space	proximity to query	class membership	clusters
Feedback	rank ordered on proximity	likelihood on class membership	relevance feedback on association value
Interaction update:			
of image of query	-	expand query	refine on the way
of features of query	refine on the way	refine on the way	alter on the way
of similarity measure	-	adapt to group	reshape to goal

Figure 1.5: Three patterns in the purpose of content-based retrieval systems.

1. Applications of Content Based Image Retrieval

We can see in Figure 1.5 that three are broad categories of user aims when using the system.

There is a broad class of methods and systems aimed at browsing through a large set of images from unspecified sources. Users of search by association at the start have no specific aim other than find interesting things. Search by association often implies iterative refinement of the search, the similarity or the examples with which the search was started. Systems in this category typically are highly interactive, where the specification may by sketch or by example images. The result of the search can be manipulated interactively by relevance feedback. To support the quest for relevant results, other sources than images are also employed.

Another class of users aims the search at a specific image. The search may be for a precise copy of the image in mind, as in searching art catalogues. Target search may also be for another image of the same object of which the user has an image. This is target search by example. Target search may also be applied when the user has a specific image in mind and the target is interactively specified as similar to a group of given examples. These systems are suited to search for stamps, art, industrial components, and catalogues, in general.

The third class of applications, category search, aims at retrieving an arbitrary image representative of a specific class. It may be the case that the user has an example and the search is for other elements of the same class. Categories may be derived from labels or emerge from the database. In category search, the user may have available a group of images and the search is for additional images of the same class. A typical application of category search is catalogues of varieties. Systems in this category are usually interactive with a domain specific definition of similarity.

These three types of use are not the whole story. A study [5] of journalists identified five typical patterns of use: searches for one specific image, general browsing to make an interactive choice, searches for a picture to go with a broad story, searches to illustrate a document, and searches for fill-ins only on the esthetic value of the picture.

II. Background Study on Image Segmentation

A. Introduction

The term image segmentation refers to the partition of an image into a set of regions that cover it. The goal in many tasks is for the regions to represent meaningful areas of the image. In other analysis tasks, the regions might be sets of border pixels grouped into such structures as line segments and circular arc segments in images of 3D industrial objects. Regions may also be defined as groups of pixels having both a border and a particular shape.

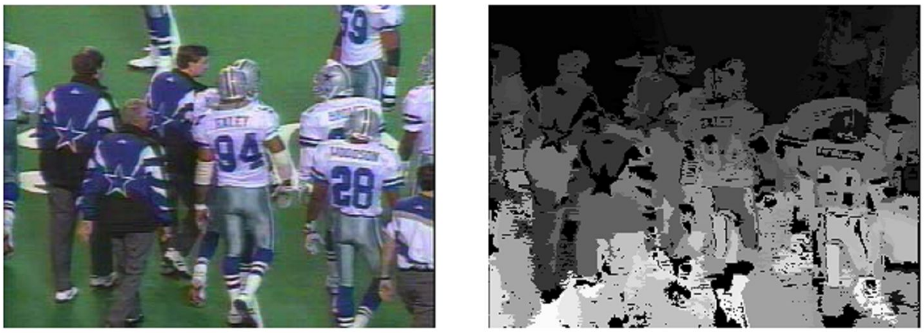

Figure 2.1: Football image (left) and segmentation into regions (right). Each region is a set of connected pixels that are similar in color.

Segmentation has two main objectives. The first objective is to decompose the image into parts for further analysis. In simple cases, the environment might be well enough controlled so that the segmentation process reliable extracts only the parts that need to be analyzed further. The segmentation is reliable, provided that the person's clothing or room background does not have the same color components as a human face. In complex cases, such as extracting a complete road network from a grayscale aerial image, the segmentation problem can be very difficult and might require application of a great deal of domain building knowledge.

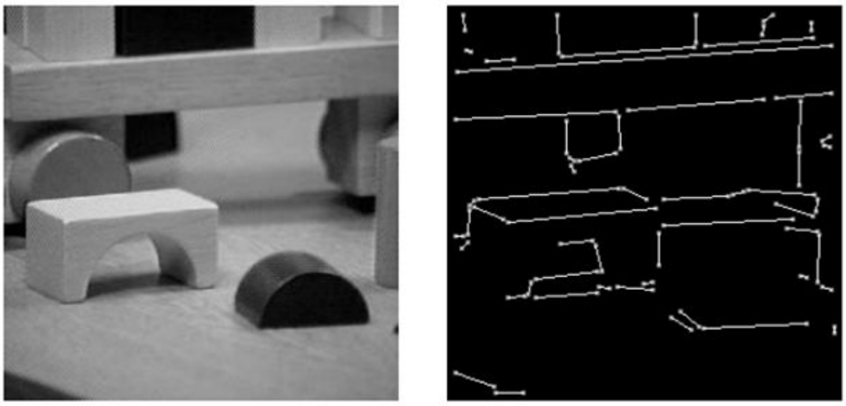

Figure 2.2: Blocks image (left) and extracted set of straight line segments (right). The line segments were extracted by the ORT (Object Recognition Toolkit) package.

The second objective of segmentation is to perform a change of representation. The pixels of the image must be organized into higher-level units that are either more meaningful or more efficient for further analysis (or both). A critical issue is whether or not segmentation can be performed for many different domains using general bottom-up methods that do not use any special domain knowledge.

B. Previous Work on Image Segmentation

It is perhaps not surprising that the earliest segmentation techniques were based on gray-level similarity. These algorithms were designed to locate simple objects, which could be assumed to project to reasonably uniform image regions in terms of image intensity. The task of the algorithm would be to identify contiguous pixels with similar gray-level value, and group them into regions. Several classifications have been proposed for algorithms that are based on gray-level pixel similarity, and the number of existing variations of these methods number well into the hundreds. Comprehensive reviews of early segmentation techniques can be found in [7], and [8]. Here we will discuss two broad classes of segmentation methods: Gray level thresholding, and region growing/merging techniques. Gray level thresholding is a generalization of binary thresholding [9]. Binary thresholding works by determining the gray level value that separates pixels in the foreground from pixels in the background, and generating a “thresholded image” where pixels are assigned one of two possible values corresponding to “foreground” and ′background′ depending on whether their gray level is above or below the selected threshold. Usually, the threshold level is determined by examining the intensity histogram for the image. If the foreground and background are simple and well differentiated, the histogram will show two large peaks corresponding to the dominant gray value in each region. The threshold level corresponds to the minimum point at the valley dividing the two peaks in the histogram.

When the object and background are not uniformly colored, or when there are many objects with different characteristics in the image, a single threshold is insufficient to perform segmentation. In general it becomes necessary to select a set of gray level thresholds that separate different peaks in the intensity histogram. Each peak is assumed to correspond to pixels that belong to a single, uniformly colored object, and all pixels within the interval defined by two consecutive threshold values are assigned the same, unique label that corresponds to a region in the image. Notice that the segmentation is imperfect even though there are three clear peaks in the intensity histogram. Objects with non-uniform color, intensity gradients caused by illumination or variations in surface reflectance, texture, and backgrounds that are not uniformly colored, complicate the selection of suitable threshold values. Moreover, it is often the case that even if suitable values can be found, the resulting segmentation is inaccurate because of overlap in gray-level intensities between different elements of the image, which leads to disconnected regions with the same label.

This leads to an important problem faced by all image segmentation algorithms: In the absence of prior knowledge about the image′s contents, it is in general not possible to determine how many regions are required for a reasonable segmentation. This problem manifests itself in two forms: Under-segmentation, which occurs when parts of the image that actually correspond to different objects, or to an object and the background, are assigned to the same region; and over-segmentation, which occurs when parts of the image corresponding to a single object are split apart.

The second class of the early segmentation algorithms mentioned above starts with a set of seed regions (individual pixels at the start of the procedure), and produces a segmentation by iteratively merging together regions that are similar enough. There are many algorithms for growing the regions, as well as for evaluating similarity between neighboring elements [7], but the fundamental principle is the same: Each initial region will grow until no more similar elements can be added to it. When none of the regions in the image can grow any more the segmentation process is complete.

The last step in the segmentation is region merging; this step uses a merge score composed of a pair-wise comparison of several region features. This process is repeated iteratively until no further merging is possible. Since the algorithm can only merge regions, the thresholds used during the local, threshold based segmentation stage are selected so that they'll yield a significantly over-segmented image; the merging step is then relied upon to turn the un-segmented image into a reasonable segmentation. Results presented in [11] show that this algorithm produces good segmentations in parts of the image that are reasonably homogeneous, and over-segmented regions when there is texture, significant intensity gradients, or objects with non-uniform coloring. The algorithm is not without problems, as there are several thresholds that must be chosen carefully depending on the image, and the region boundaries themselves have slight artifacts introduced by the sector-based initial segmentation. Even so, the algorithm illustrates what can be achieved with thresholding/merging schemes.

III. Color Space

A. Introduction

Describing colors accurately is a matter of great commercial importance. Many products are closely associated with very specific colours . for example, the golden arches; the color of various popular computers; the color of photographic film boxes and manufacturers are willing to go to a great deal of trouble to ensure that different batches have the same color. This requires a standard system for talking about color. Simple names are insufficient, because relatively few people know many color names, and most people are willing to associate a large variety of colors with a given name.

Color matching data yields simple and highly effective linear color spaces. Specific applications may require color spaces that emphasize particular properties or uniform color spaces, which capture the significance of color differences.

B. Linear Color Space

There is a natural mechanism for representing color: first, agree on a standard set of primaries, and then describe any colored light by the three values of the weights that people would use to match the light using those primaries. In principle, this is easy to use . to describe a color, we set up and perform the matching experiment and transmit the match weights.

Performing a matching experiment each time we wish to describe a color can be practical. For example, this is the technique used by paint stores; you take in a flake of paint, and they'll mix paint, adjusting the mixture until a color match is obtained. Paint stores do this because complicated scattering effects within paints mean that predicting the color of a mixture can be quite difficult. However, Grassman's laws mean that mixtures of colored lights - at least those seen in a simple display – mix linearly, which means that a much simpler procedure is available. RGB is the an example of linear color space and it is explained below:

1. RGB (Red, Green and Blue) Color Space

Colour spaces are normally invented for practical reasons, and so a wide variety exist. The RGB colour space is a linear colour space that formally uses single wavelength primaries (645.16 nm for R, 526.32nm for G and 444.44nm for B). Informally, RGB uses whatever phosphors a monitor has as primaries. Available colours are usually represented as a unit cube – usually called the RGB cube – whose edges represent the R, G, and B weights. The cube is drawn in Figure 2.5; remember, since the weights are the weights associated with primary lights, red and green mix to give yellow.

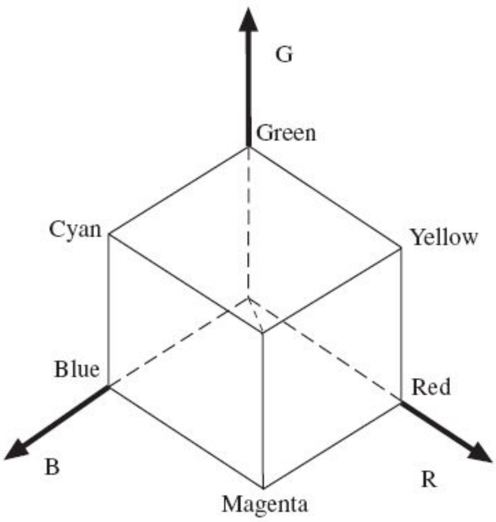

Figure 3.1: RGB Cube

C. Non-Linear Color Spaces

The coordinates of a colour in a linear space may not necessarily encode properties that are common in language or are important in applications. Useful colour terms include: hue – the property of a colour that varies in passing from red to green saturation – the property of a colour that varies in passing from red to pink; and brightness (sometimes called lightness or value) – the property that varies in passing from black to white. For example, if we are interested in checking whether a colour lies in a particular range of reds, we might wish to encode the hue of the colour directly.

Another difficulty with linear colour spaces is that the individual coordinates do not capture human intuitions about the topology of colours; it is a common intuition that hues form a circle, in the sense that hue changes from red, through orange to yellow and then green and from there to cyan, blue, purple and then red again. Another way to think of this is to think of local hue relations: red is next to purple and orange; orange is next to red and yellow; yellow is next to orange and green; green is next to yellow and cyan; cyan is next to green and blue; blue is next to cyan and purple; and purple is next to blue and red. Each of these local relations works, and globally they can be modelled by laying hues out in a circle. This means that no individual coordinate of a linear colour space can model hue, because that coordinate has a maximum value which is far away from the minimum value.

1. HSV (Hue, Saturation and Value) Color Space

HSV color space is obtained by looking down the center axis of the RGB cube. Because RGB is a linear space, brightness – called value in HSV – varies with scale out from the origin, and we can “flatten” the RGB cube to get a 2D space of constant value, and for neatness deform it to be a hexagon. This gets the structure shown in Figure 2.6, where hue is given by an angle that changes as one goes round the neutral point and saturation changes as one moves away from the neutral point.

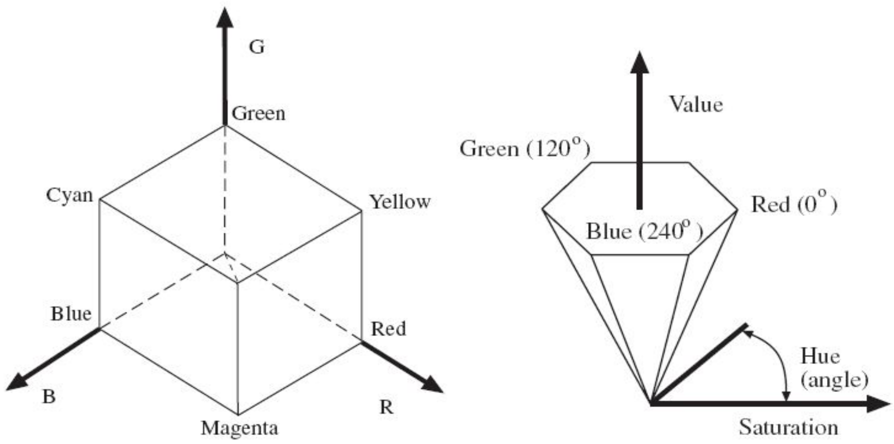

Figure 3.2: On the left, we see the RGB cube; this is the space of all colours that can be obtained by combining three primaries (R, G, and B - usually defined by the colour response of a monitor) with weights between zero and one. It is common to view this cube along its neutral axis - the axis from the origin to the point (1, 1, 1) - to see a hexagon, shown in the middle. This hexagon codes hue (the property that changes as a colour is changed from green to red) as an angle, which is intuitively satisfying. On the right, we see a cone obtained from this cross-section, where the distance along a generator of the cone gives the value (or brightness) of the colour, angle around the cone gives the hue and distance out gives the saturation of the colour.

D. Conversion from RGB to HSV Color Space

Being closer to human perception HSV color space was chosen instead of RGB color space. To perform the adaptive segmentation on the image to extract the feature vector, first input image have to be converted from RGB color space to HSV color space. Following equations are used for conversion from RGB color space, which is the most general method to express color, to HSV color space.

$$h = (cos)*\{[((R-G)+(R-B))/2]*[(R-G)*2+(R-G)*(G-B)]*(1/2)\} \tag{3.1}$$

$$H = h \quad \text{where } G > B \tag{3.2}$$

$$H = 2*PI - h \quad \text{otherwise} \tag{3.3}$$

$$V = (R+G+B)/3 \tag{3.4}$$

$$S = 1- \min(R,\ G,\ B)/V \tag{3.5}$$

Where, R, G and B represent red, green and blue components respectively with values between 0-255. H stands for hue, V stands for value and S stands for saturation.

For the purpose of our algorithms we used the range 0-1 for Saturation and Value components of HSV information space and 0^o-360^o for Hue. For this purpose we used following simple equations:

$$H = ((H/255)*360)\bmod 360 \tag{3.6}$$

$$V = V/255 \tag{3.7}$$

$$S = S/255 \tag{3.8}$$

After using these equations the hue will rang from 0^o to 360^o, saturation and value will range from 0 to 1. This will make the computation easier.

IV. Proposed Adaptive Segmentation Techniques

A. Introduction

This dissertation is actually a compiled form of our published papers in different international and local conferences and Journals. Methods which we have proposed are based on adaptive segmentation of HSV cone. Different techniques are used to adaptively segment the image by using image's pixel information. our algorithms are based on color image segmentation. Following four main colored based adaptive segmentation techniques are explained in this dissertation.

- Efficient Image Retrieval using Adaptive Segmentation of HSV Color Space.
- Enhanced Segmentation Technique for Image Retrieval.
- Feature Vector Extraction System based on Adaptive Segmentation of HSV Information Space.
- Extracting Color Using Adaptive Segmentation for Image Retrieval.

Following figures, Figure 4.1, 4.2 and 4.3, show the general simulation systems of the proposed algorithms. the simulation engine consists of feature extraction process, batch feature extraction and storage process for a collection of images, and the interactive retrieval process.

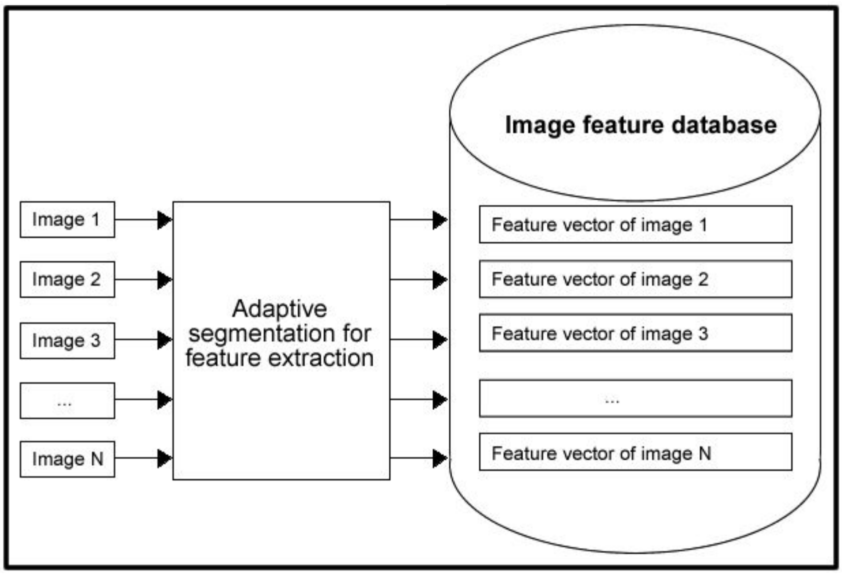

Figure 4.1: Feature extraction and storage process for an image collection

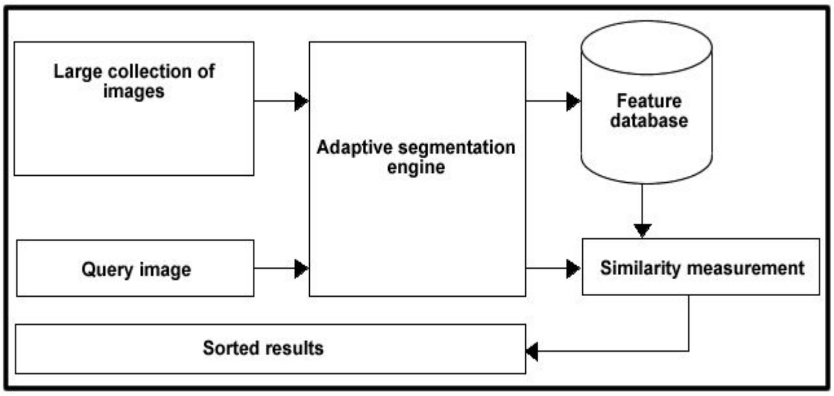

Figure 4.2: Overview of entire retrieval process

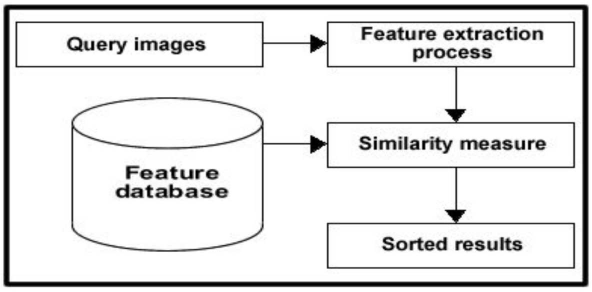

Figure 4.3: Query by image content process

B. Image Retrieval Related Work

Most important elements in a content based retrieval system are the features used to express an image and image features can be classified into three levels. First, primitive features include color, shape and texture information of an image. Second, the logical features contain the logical information for recognition of an object. Finally, abstract information refers to abstract expressions like feeling, emotion and importance of a scene. Most research on retrieval technology being carried out currently uses primitive features like color, shape and texture as image information.

Technology based on color tone is most widely used due to compactness of calculation and information expression. Histogram is mainly used for color tone information. Representative work is the Histogram Intersection method proposed by Swain [11]. Extended Swain method which compares color tone histogram to recognize objects [12], and Distance Method and Reference Color Table method [13] for comparing color tone were also proposed. Methods using color tone is robust with respect to object movement, rotation and to changes like distortion within an image and may be implemented easily. However, performance will decrease significantly for changes in lighting. There is also the possibility that unwanted image with similar color tone distribution may be the output [14].

Image retrieval using shape information is the most difficult method to implement. In order to use shape information, separation of objects within an image is important. Therefore, best performance is obtained for cases in which object contour is clear [14]. Once contour of an object is determined, its shape can be expressed with information including area, eccentricity, circularity, form signature, curvature and fractal dimension. Retrieval techniques using shape information that have been proposed include those that use the following feature parameters: Chain Code [15], Zernike Moment [16], Invariant Moment [15] and Fourier Descriptor [17].

Image retrieval using texture information has been advanced along with multiple resolution filtering techniques like the Wavelet Transform. Texture is a pattern in which many visual elements are concentrated and uniformly arranged [18]. To analyze this, much research has been carried out on using multi-directional filter banks and on model based analysis using Markov models [18]. Recently, a method that decomposes and analyzes image texture information for each frequency band, using the Gabor wavelet transformation, has become popular [19].

In addition, methods that combine two or more features to use as information for retrieval have also been proposed [20]. J. Huang [20] has imported Color Correlogram and simultaneously considered color tone information and spatial assignment information for color tone. G. Pass and R. Zabih [21] proposed a method which uses regional information obtained through refinement of each color tone bin for the color tone histogram.

Content based image retrieval techniques that have been developed into commercial products include QBIC of IBM, Virage VIR Image Engine by Virage and Image Retrieval Ware from Excalibur. In addition, many experimental demo systems exist on WWW. Representative works include Photobook from MIT, Columbia University's WebSEEk and Carnegie-Mellon University's Informedia. Content based image retrieval systems have started to enter the market and key application areas include crime prevention (finger print recognition and face recognition), intellectual property protection (trade mark registration), broadcast media and advertisements (video material management) and web search. Retrieval engines used by representative web search engines AltaVista and Yahoo provide retrieval functions from Virage and Excalibur, respectively.

C. Image Databases Used for Experimentations

In order to test the algorithm proposed in this paper, a retrieval system was implemented on the web. Experiments are performed by using two different image databases, one is a professor wang's database [28, 29] it contains 1000 images. The other database contains 11 sets of color images - downloaded from the website of Efficient Content-based Retrieval Group, University of Washington -

taken in different parts of the world like Australia, Iran, Italy, Barcelona, Cambridge, Japan etc. Experiments show that even if we rotate or reduce the size of the image the results are efficient. Figure 4.1 and Figure 4.2 show some sample images from both of the image databases.

Figure 4.4: Professor Wang's Sample Image Database

Figure 4.5: University of Washington's Sample Image Database

D. Efficient Image Retrieval using Adaptive Segmentation of HSV Color Space

This technique presents an efficient image retrieval system using adaptive segmentation of Hue, Saturation and Value (HSV) color space. The image is first classified into n number of areas based on different selected ranges of hue and value, then each area is partitioned into m number of segments based on the number of pixels it contains, the area which has more pixels will be partitioned into more segments and the area which has less pixels will be partitioned into less number of segments. This is used as a feature vector. Retrieval system outputs the image with a high matching factor. A small demonstration system has been tested and shows superior performance compared with the simple color based retrieval systems.

1. Adaptive Segmentation of HSV Color Space

Being closer to human perception, HSV color space is selected for adaptive segmentation instead of RGB color space. To perform the adaptive segmentation on the image for extracting feature vector, input image is converted from RGB color space to HSV color space. For this purpose we used a most general method, which is explained in section D of chapter III.

After using these equations the hue will rang from 0^o to 360^o, saturation and value will range from 0 to 1. Since we know hue represents the angle that is why we have converted the value of hue from 0-255 to 0^o-360^o. This will make the computation easier.

Figure 4.6: RGB image (left), HSV image (right)

After converting the RGB image to HSV image, *n* different areas are identified based on different ranges of hue and value. Following table shows the different selected ranges of hue and value which we used in our demo system.

Table 4.1: Different Ranges of Hue and Values used in the experiment

S#	Hue	Value
1	0<H<=60	0<V<=0.5
2		0.5<V<=0.75
3		0.75<V<=1.0
4	60<H<=120	0<V<=0.5
5		0.5<V<=0.75
6		0.75<V<=1.0
7	120<H<=180	0<V<=0.5
8		0.5<V<=0.75
9		0.75<V<=1.0
10	180<H<=240	0<V<=0.5
11		0.5<V<=0.75
12		0.75<V<=1.0
13	240<H<=300	0<V<=0.5
14		0.5<V<=0.75
15		0.75<V<=1.0
16	300<H<=360	0<V<=0.5
17		0.5<V<=0.75
18		0.75<V<=1.0

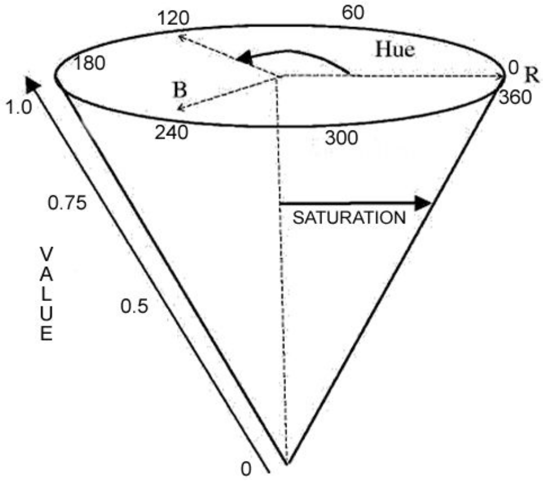

Figure 4.7: HSV Color Space diagram displays adaptive segmentation. The area which has more pixels is partitioned into more number of segments and the area which has less pixels is partitioned into less number of segments.

HSV color tone space is partitioned into n number of areas, by using different ranges of hue and value, and then each area is further partitioned into m number of segments depending upon the number of pixels in that area, the areas which has more pixels is partitioned into more number of segments and the areas which have less pixels is divided into less number of segments as shown in Figure 4.4. Through this procedure distribution characteristics for the color tone distributed in the corresponding image is extracted. Such operations are carried out for the entire image and then feature vector is computed.

Following equation is used to partition each area in m number of segments; m can have different values for different areas.

$$m_i = (X_i/T)*S \qquad 0<i<n \tag{4.1}$$

Where m_i represents the segments in area i, Xi represents the number of pixels in area i (where i reaches from 0 to n), T represents total number of pixels of the image and S represents total number of segments of the HSV image.

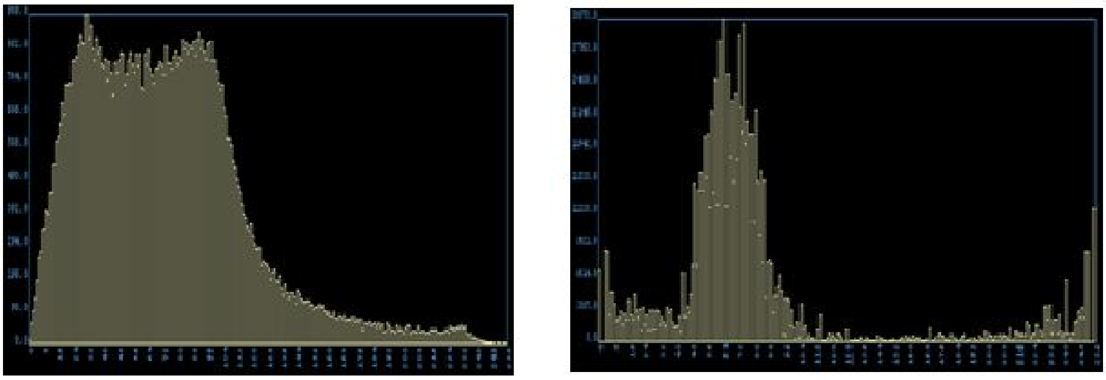

Figure 4.8: value histogram of Figure 4.3 (left), hue histogram of Fig 4.3 (right).

From the above Figure 4.8 we can clearly see that most of the pixels of the image are between $67^{o}-124^{o}$ for hue range and 0-0.5 for value range. When we will perform the adaptive segmentation on this image the maximum segments will be found in area which will have range of hue between $67^{o}-124^{o}$ and value between 0-0.5.

Table 4.2 contains the results after performing the adaptive segmentation on the sample image Figure 4.3 (right). We can clearly see from the table that the area which has more pixels is partitioned into more number of segments.

Figure 4.6 represents the complete flowchart for the image retrieval system implemented for demonstration of the algorithm proposed in this paper.

Table 4.2: Different number of segments in different areas.

S#	Hue	Value	Segments
1	0<H<=60	0<V<=0.5	5
2		0.5<V<=0.75	1
3		0.75<V<=1.0	1
4	60<H<=120	0<V<=0.5	23
5		0.5<V<=0.75	1
6		0.75<V<=1.0	1
7	120<H<=1180	0<V<=0.5	5
8		0.5<V<=0.75	1
9		0.75<V<=1.0	1
10	180<H<=240	0<V<=0.5	1
11		0.5<V<=0.75	1
12		0.75<V<=1.0	1
13	240<H<=300	0<V<=0.5	1
14		0.5<V<=0.75	1
15		0.75<V<=1.0	1
16	300<H<=360	0<V<=0.5	2
17		0.5<V<=0.75	2
18		0.75<V<=1.0	1

From flow chart, Figure 4.9, we can see that initially the query image is loaded in the memory. After loading the reference query image, it is converted to HSV color space using the equations as mentioned earlier. Then the adaptive segmentation is performed in which HSV color space is partitioned into m number of segments and it is used as a feature vector. This feature vector is used for comparison with the images in the database and matching for each image is calculated using Euclidean distance equation:

$$f(t) = \sum_{i=1}^{n} |Q_i - I_i| \quad 1<i<n, \tag{4.2}$$

Where, Q_i is the query image segments in area i and I_i is the comparison image segments in area i where i reaches from 1 to n.

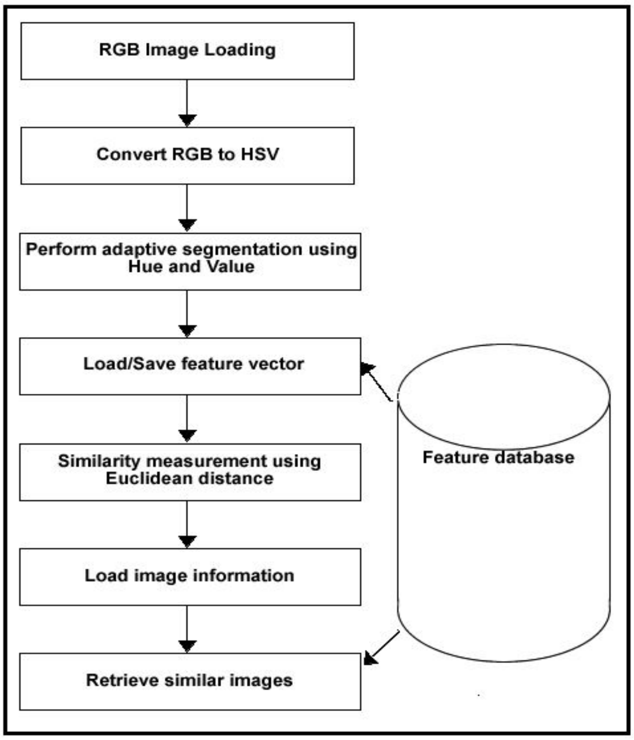

Figure 4.9: Flowchart of efficient image retrieval using adaptive segmentation of HSV color space

When matching calculation is completed, results are sorted in ascending order and the image which resembles more with the query image is brought to the top.

In order to test the algorithm proposed in this paper, a simple retrieval system was implemented on the web.

Experiments show that even if we rotate or reduce the size of the image the results are accurate.

Figure 4.10 shows the experiment results, when we executed the query on image database which contains 1000 images. We can see from different results that the first image retrieved is always the query image, since the query image is also the part of database images. Experiments also show that if we change the size of the image or rotate the image the results are even then good.

We used an image database of 450 different images using which we got the results which were shown in Figure 4.11. In this image database some of the images were changed in size and rotated (90° and 180°). As shown in the experiment results, retrieval is carried out for the corresponding image even when the image is rotated or the size is changed.

Figure 4.10: Experiment Results

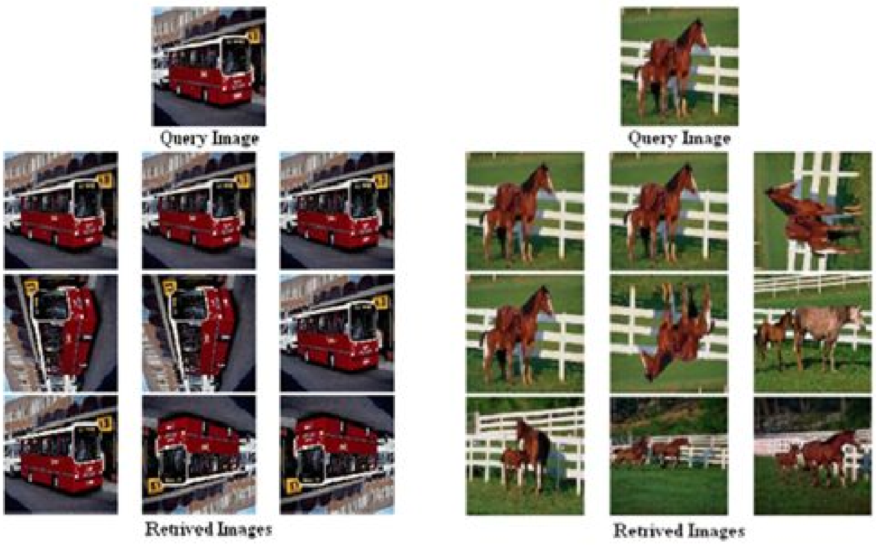

Figure 4.11: Experiment results. Image is successfully retrieved even if it is rotated or changed in size

From Figure 4.12 we can see that some of the images retrieved are irrelevant images, which means that our algorithm requires some refinements. In future we will work to refine our algorithm to improve its performance and accuracy.

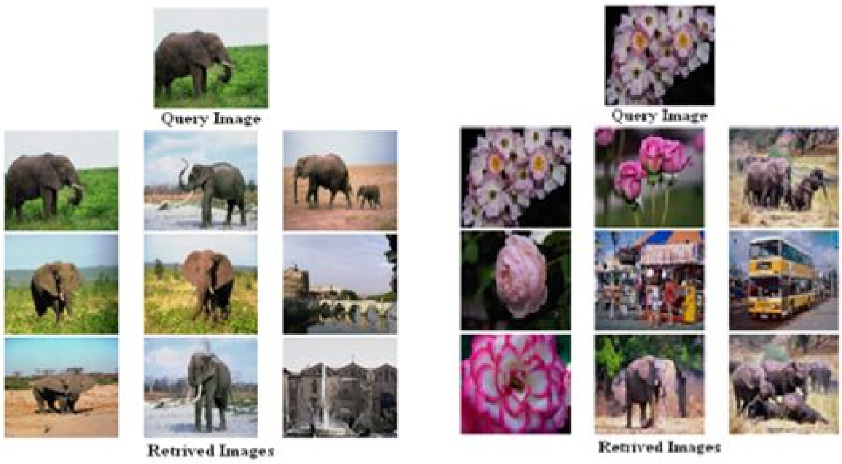

Figure 4.12: Experiment results, not very good results.

E. Enhanced Segmentation Technique for Image Retrieval

Next feature extraction technique for image retrieval is based on color features of the image. The proposed method extracts the features by using a novel technique for segmentation of image based on different ranges of Hue and Saturation in HSV color space. Image is first classified or divided into AR number of areas, and then each area is partitioned into S number of segments based on the number of color pixels it contains. A web-based retrieval system is used for experimentation and results.

1. Pre-Processing

HSV is widely used in computer graphics. The RGB coordinates can be easily translated to the HSV color space. The three color characteristics, hue, saturation, and value or lightness, are defined to distinguish color components. Hue describes the actual wave length of the color by representing the color name, for example, red, blue, or yellow. Saturation is a measure of purity of a color. It indicates how much white light is added to a pure color. For instance the color red is a 100% saturated color, but pink is a low saturated color because of the amount of white in the color. Lightness embodies the intensity of the color. It ranges from black to white. The use of a color′ hue and saturation is defined as the chromaticity of a color.

Using spaces such as HSV, there is the advantage of utilizing their more intuitive properties, notably Hue. The hue component, however, contains irrelevant values where the saturation or chroma is very low or zero and as such Hue histograms can contain erroneous data.

As we know that the HSV color space is closer to human perception, so we used HSV instead of RGB color space. To perform the segmentation on the imagefor extracting color features, we first must convert the input image from RGB color space to HSV color space. For this purpose we can use a method explain in Section D of chapter Ⅲ.

2. Enhanced Segmentation

After conversion from RGB to HSV the next design parameter is the quantization of the color space. The HSV color space is cylindrical. The long axis represents value: blackness to whiteness. Distance from the axis represents saturation: amount of color present. The angle around the axis is the hue: tint or tone. Since hue represents the most significant characteristic of the color, it requires the most fine quantization. In the hue circle the primaries red, green and blue are separated by 120 degrees. A circular quantization at 20 degree steps sufficiently separates the hues such that the three primaries and yellow, magenta and cyan are represented each with three sub-divisions.

In term of this information, Bodi [27] quantized HSV color space, 19 hues step

20 (0, 20, 40, ⋯, 360), 4 saturations, step 0.25 (0.173, 0.423, 0.673, 0.923) and 5 values, step 0.25 (0, 0.25, 0.5, 0.75, 1.00). We have tested a large number of images, and made a slight modification. We used only hue and saturation and no value, we define: 12 hues, step 30 and 5 saturations, step 0.2 as shown in table 4.1.

As a result we get 12*5 = 60 number of different areas each represent different color distribution of the image. Table 1 shows the different selected ranges of hue and saturation which we used in our demo system.

After partitioning into AR number of areas, by using different ranges of hue and saturation, as shown above, each area is further divided into S number of segments depending upon the number of pixels in that area, the areas which has more pixels are partitioned into more number of segments and the areas which has less pixels are divided into less number of segments. Through this procedure we are able to get the distribution characteristics for the color tone distributed in the corresponding image. Such operations are carried out for the entire image and then feature vector is computed.

Table 4.3: Different Ranges of Hue and Saturation used in the experiment

S#	Hue	Saturation
1	0<H<=30	0<S<0.2
2		0.2<=S<0.4
3		0.4<=S<0.6
4		0.6<=S<0.8
5		0.8<=S<1.0
6	30<H<=60	0<S<0.2
7		0.2<=S<0.4
8		0.4<=S<0.6
9		0.6<=S<0.8
10		0.8<=S<1.0
...	...	...
60	330<H<=360	0.8<=S<1.0

Following equation is used to partition each area in S number of segments; S can have different values for different areas.

$$S_i = (X_i/T)*TS \qquad 0<i<AR \tag{4.3}$$

Where S_i represents the number of segments in area i, X_i represents the number of pixels in area i (where i ranges from 1 to AR), T represents total number of pixels of the image and TS represents total number of required segments of the entire HSV image.

Figure 4.13 shows the complete flowchart for the image retrieval system implemented for demonstration of the algorithm proposed in this paper.

Initially the query image is loaded in the memory. After loading the reference query image, it is converted to HSV color space using the equations as mentioned earlier. Then the segmentation is performed in which HSV color space is partitioned into S number of segments and it is used as a feature vector. This feature vector is used for comparison with the images in the database and matching for each image is calculated using Euclidean distance equation. Segments in area of the query image are compared with the corresponding area of the database image as elaborated in the equation below:

$$f(t) = \sum_{i=1}^{n} \left| Q_i - I_i \right| \qquad 1<i<AR, \tag{4.4}$$

Where, Q_i is the query image segments in area i and I_i is the comparison image segments in area i where i ranges from 1 to m.

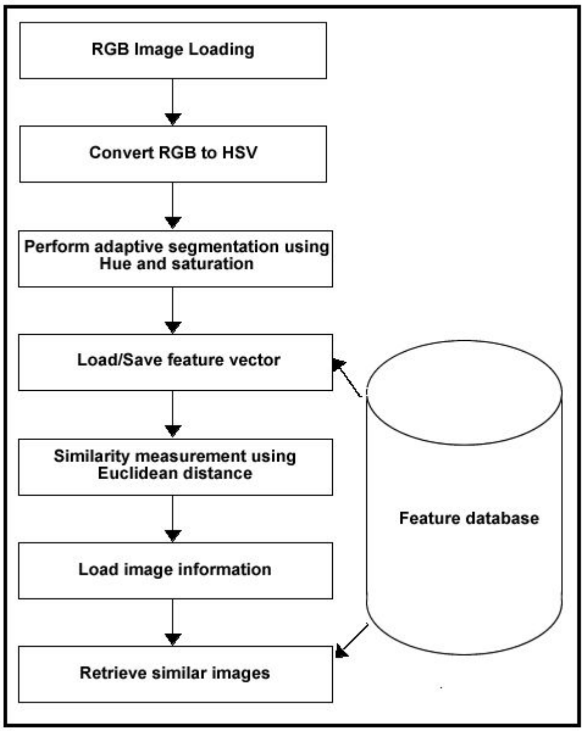

Figure 4.13: Flowchart of enhanced segmentation technique for image retrieval

3. Experiment and Results of Enhanced Segmentation Technique for Image Retrieval

To test the algorithm presented in this paper, we developed a testing web application. By using the application we tested the algorithm on database provided by Professor Wang [28, 29]. This database contains different categories of images like dinosaurs, horses, buses, human life, etc.

Experiments show that even if we rotate or reduce the size of the image the images are still retrieved accurately.

Figure 4.14 shows the experiment results, when we executed the query on image database which contains 1000 images. We can see from different results that the first image retrieved is always the query image, since the query image is also the part of database images. Experiments also show that if we change the size of the image or rotate the image the results are even then better. Figure 4.15 shows results, when we executed the query on image database which contains 450 images and some of those images were rotated (90 or 180 degrees) and changed in size (25% or 50%).

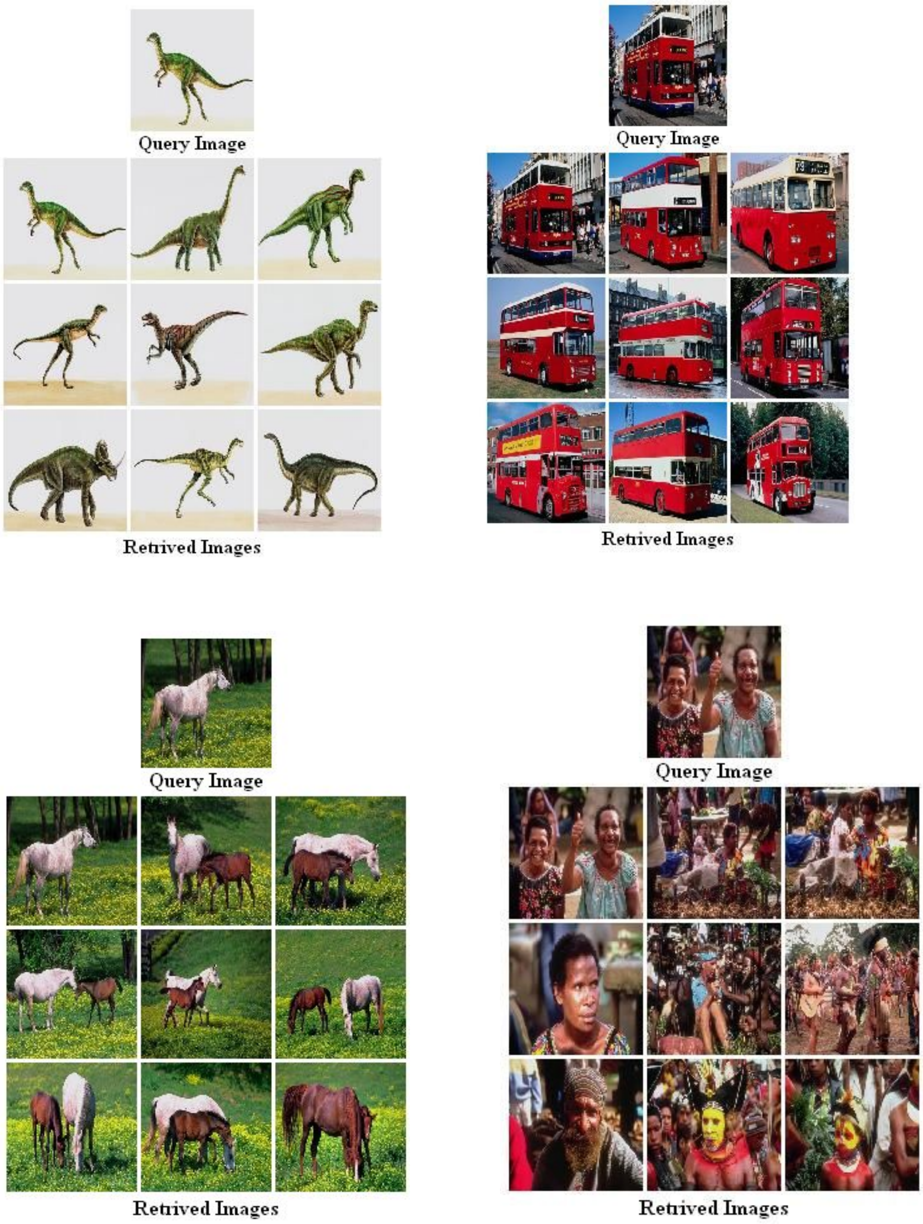

Figure 4.14: Experiment results of enhanced segmentation technique for image

retrieval

From Figure 4.16 we can see that some of the images retrieved are irrelevant images, which means that our algorithm requires some refinements. In future we will work to refine our algorithm to improve its performance and accuracy.

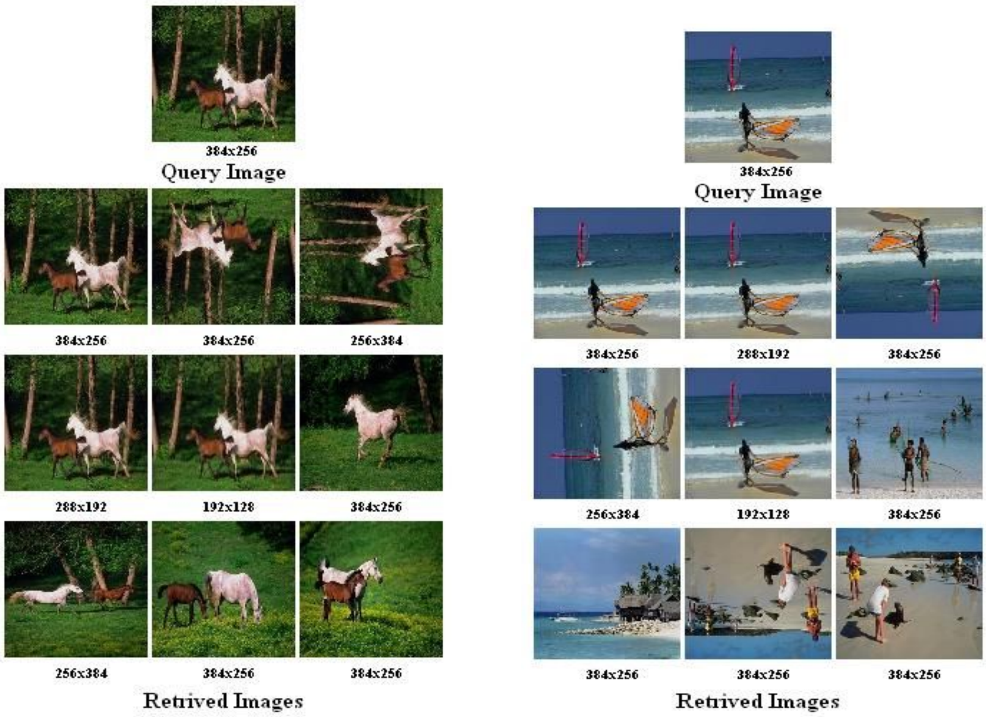

Figure 4.15: Experiment results of enhanced segmentation technique for image retrieval

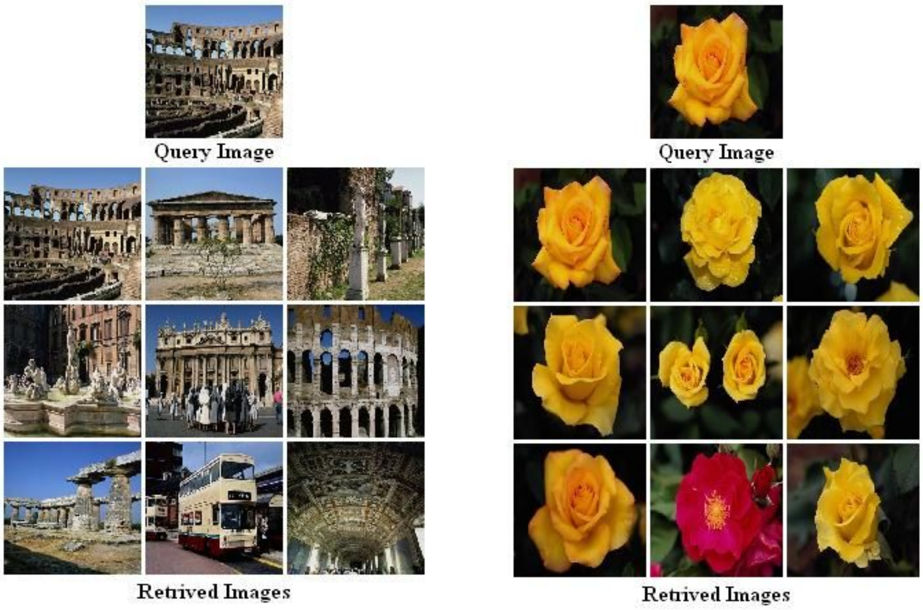

Figure 4.16: Experiment results of enhanced segmentation technique for image retrieval

F. Feature Vector Extraction System Based On Adaptive Segmentation of HSV Information Space

Color is a rich and complex experience, usually caused by the vision system responding differently to different wavelengths of light (other causes include pressure on the eyeball and dreams). Color of object can play an important role in recognizing that object from the image. Different kinds of color spaces have been established and studied in the past for image retrieval. In this paper also we have studied HSV color space. We used a feature extraction technique based on the adaptive segmentation of the image using its color information. By using different ranges of Hue, Saturation and Value, image is first classified into different number of areas, and then each area is partitioned into different number of segments. Our focus is in the domain of photographic images with an essentially unlimited range of topics.

1. Proposed Algorithm

The first step for extracting the feature vector is same as explained in section 3 of chapter ii i.e. convert RGB image to HSV color space. After conversion from RGB to HSV the next design parameter is the quantization of the color space.

Bodi [27] quantized HSV color space, 19 hues step 20 (0, 20, 40, ⋯, 360), 4 saturations, step 0.25 (0.173, 0.423, 0.673, 0.923) and 5 values, step 0.25 (0, 0.25, 0.5, 0.75, 1.00). We have tested a large number of images, and made a slight modification. We used only hue and saturation and no value, we define: 18 hues, step 20 (350, 10, 30, 50⋯ 350), 5 saturations, step 0.2 (0, 0.2, 0.4, 0.6, 0.8, 1.0), and 3 values, step 0.25 (0, 0.5, 0.75, 1.0) as shown in table 1.

As a result we get 18*5*3 = 270 number of different areas each represent different color distribution of the image. Following table shows the different selected ranges of hue, saturation and value which we used in our demo system.

We can see from the table 1 that initially the ranges of Hue and saturation are kept constant but different ranges of value are used. In the 4th iteration the value of saturation is changed. These iterations are continued until we get the pixels information from each area.

After partitioning into n number of areas, the image is further divided into m number of segments depending upon the number of pixels in that area, the areas which have more pixels are partitioned into more number of segments and the areas which have fewer pixels are divided into less number of segments. Through this procedure we are able to get the distribution characteristics for the color tone in the corresponding image. Such operations are carried out for the entire image and then feature vector is computed, the size of the feature vector depends upon the number of areas chosen in the experiment as in our case it will be 270.

Table 4.4: Different ranges of hue, saturation and value used in the experiment

S#	Hue	Saturation	Value
1			0<V<=0.5
2		0<=S<0.2	0.5<V<=0.75
3			V>=0.75
4			0<V<=0.5
5		0.2<=S<0.4	0.5<V<=0.75
6			V>=0.75
7			0<V<=0.5
8	350<H<=10	0.4<=S<0.6	0.5<V<=0.75
9			V>=0.75
10			0<V<=0.5
11		0.6<=S<0.8	0.5<V<=0.75
12			V>=0.75
13			0<V<=0.5
14		S>=0.8	0.5<V<=0.75
15			V>=0.75
...	...	...	...
270	330<H<=350	S>=0.8	V>=0.75

Following equation is used to partition each area in m number of segments m can have different values for different areas.

$$m_i = (X_i / T) * TS \qquad 0<i<=n \tag{5.1}$$

Where m_i represents the segments in area i, X_i represents the number of pixels in area i (where i ranges from 1 to n), T represents total number of pixels of the image and TS represents total number of required segments of the entire HSV image.

This feature vector, which we computed using the above equation, is used for comparison with the images in the database and matching for each image is calculated using Euclidean distance equation:

$$f(t) = \sum_{i=1}^{n} |Q_i - I_i| \quad 1<i<m, \tag{5.2}$$

Where, Q_i is the query image segments in area i and I_i is the comparison image segments in area i where i ranges from 1 to m.

When matching calculation is completed, results are sorted in ascending order and the image which resembles more with the query image is brought to the top.

2. Experiment and Results of Feature Vector Extraction System Based on Adaptive Segmentation of HSV Information Space

In order to test the algorithm proposed in this paper, a retrieval system was implemented on the web. Experiments are performed by using two different image databases, one is a professor wang's database [28, 29] it contains 1000 images. The other database contains 11 sets of color images - downloaded from the website of Efficient Content-based Retrieval Group, University of Washington - taken in different parts of the world like Australia, Iran, Italy, Barcelona, Cambridge, Japan etc. Experiments show that even if we rotate or reduce the size of the image the results are efficient.

Figure 4.17 shows the experiment results, when we executed the query on image database which contains 11 sets of color images downloaded from the website of Efficient Content-based Retrieval Group, University of Washington.

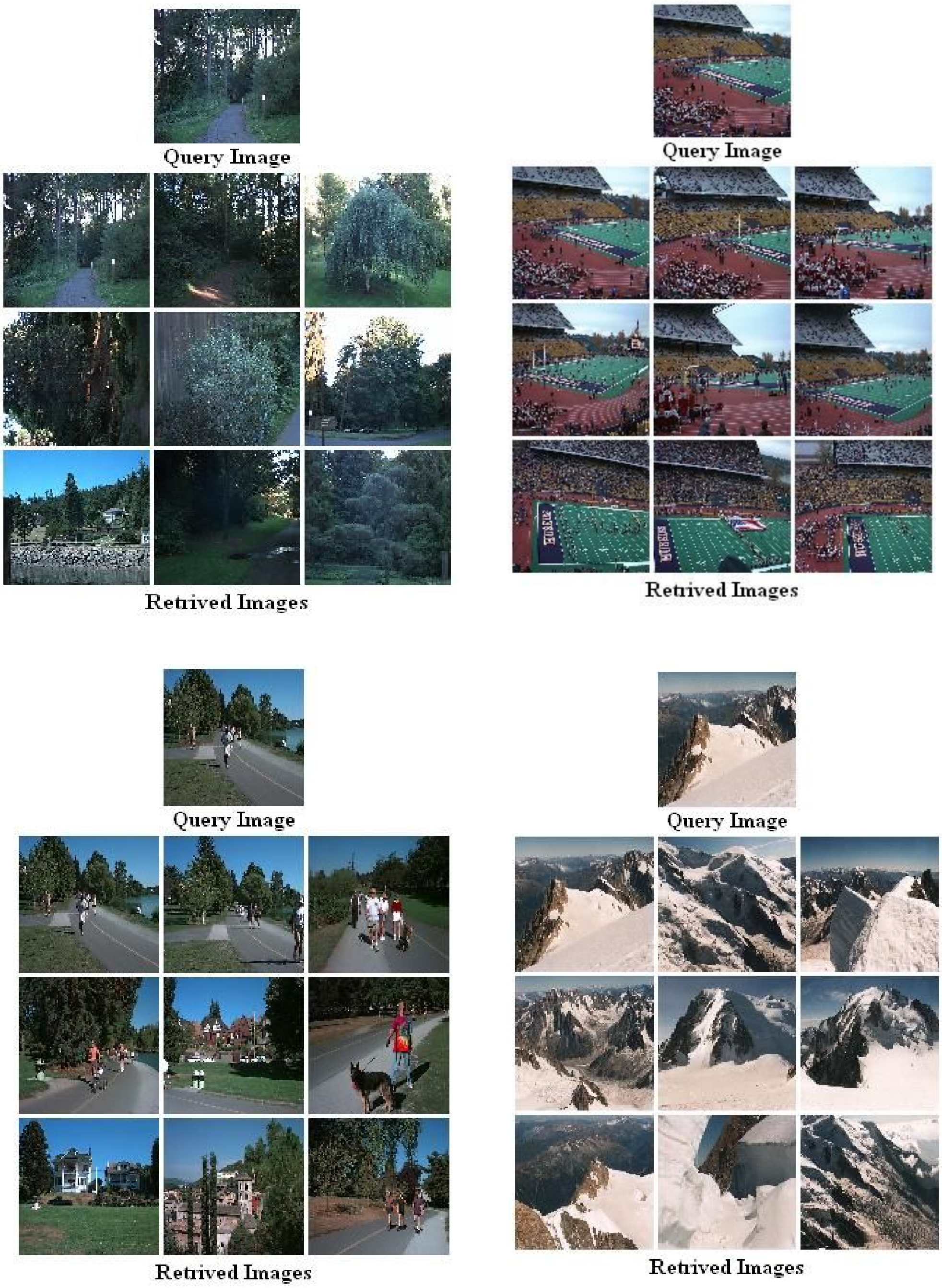

Figure 4.17: Experiment result for database of University of Washington

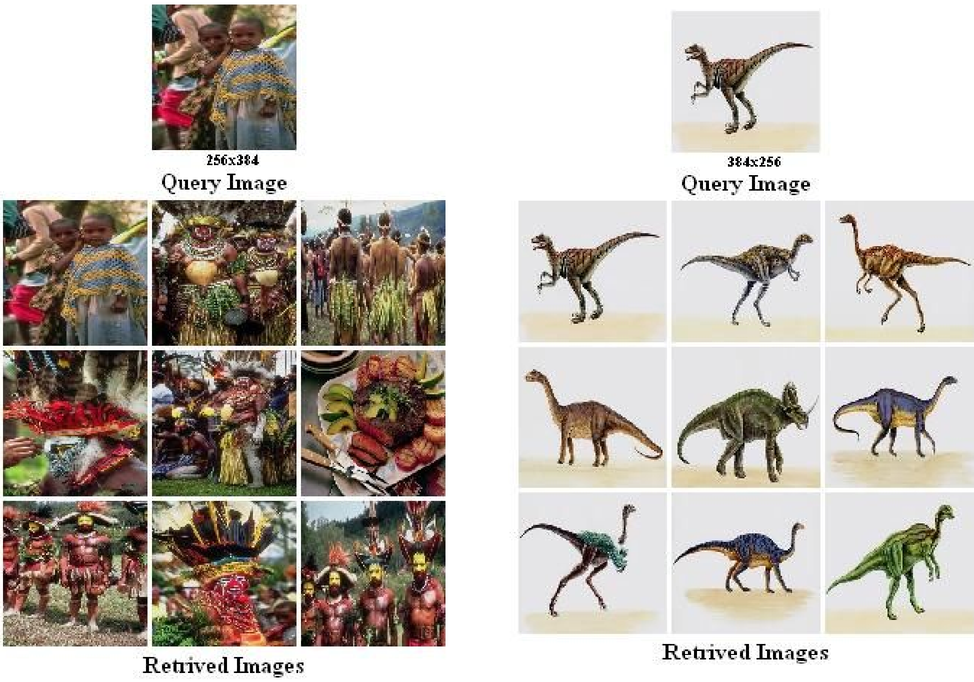

Figure 4.18: Experiment results using database of Professor Wang.

Experiments also show that if we change the size of the image or rotate the image the results are even then better. Figure 4.19 shows results, when we executed the query on image database which contains 450 images and some of those images were rotated (90 or 180 degrees) and changed in size (25% or 50%). We can clearly see from Figure 4.19 that the proposed algorithm tried to find the rotated and changed in size images of a girl running on a beach. The database contained 5 such images and all of those are retrieved accurately and efficiently.

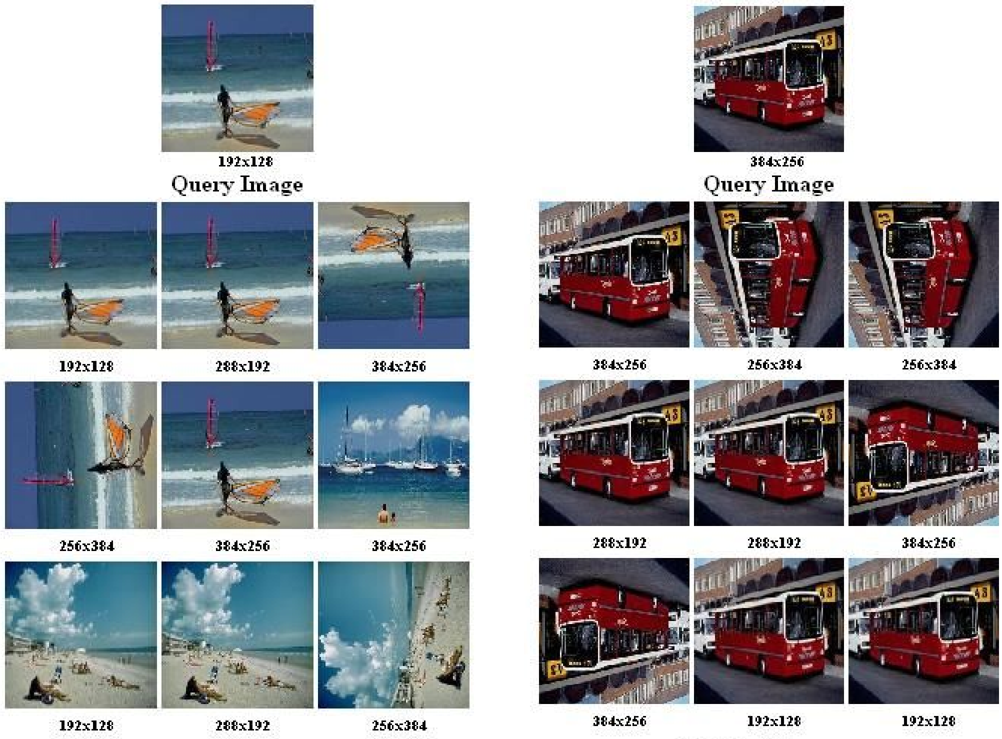

Figure 4.19: Experiment results using database of rotated and different size images.

G. Extracting Color Using Adaptive Segmentation for Image Retrieval

In this approach the issue of image database retrieval based on color using HSV cone is addressed. Histogram search characterizes an image by its color distribution but the drawback of a global histogram representation is that information about object location, shape, and texture is discarded. Thus we used local histogram to extract the maximum color occurrence from each segment. Before extracting the maximum color from each segment the input image is converted to HSV and adaptive segmentation is applied on the HSV color space. This will compute the feature vector. Different quantization of hue, saturation and value are used. Minkowski metric is used for feature vector comparison. Web based image retrieval demo system is built to make it easy to test the retrieval performance and to expedite further algorithm investigation.

1. Proposed Algorithm

After conversion from RGB to HSV color space, using the equations mentioned in section 3 of chapter iii, the next design parameter is the quantization of the color space. Since hue represents the most significant characteristic of the color, it requires the finest quantization. In the hue circle the primaries red, green and blue are separated by 120 degrees. A circular quantization at 20 degree steps sufficiently separates the hues such that the three primaries (Red, Green and Blue) and yellow, magenta and cyan are represented each with three sub-divisions.

In term of this information, Bodi [27] quantized HSV color space, 19 hues step 20 (0, 20, 40, …, 360), 4 saturations, step 0.25 (0.173, 0.423, 0.673, 0.923) and 5 values, step 0.25 (0, 0.25, 0.5, 0.75, 1.00). We have tested a large number of images, and made a slight modification. We define: 18 hues, step 20 (350, 10, 30, 50… 350), 5 saturations, step 0.2 (0, 0.2, 0.4, 0.6, 0.8, 1.0) and 3 values, steps 0.25 (0, 0.5, 0.75, 1.0).

As a result we get 18*5*3 = 270 number of different areas each represent different color distribution of the image. Table 6.1 shows the different selected ranges of hue, saturation and value.

Table 4.5: Different ranges of Hue, Saturation and Value used in the experiment

S#	Hue	Saturation	Value
1	350<H<=10	0<=S<0.2	0<V<=0.5
2			0.5<V<=0.75
3			V>=0.75
4		0.2<=S<0.4	0<V<=0.5
5			0.5<V<=0.75
6			V>=0.75
7		0.4<=S<0.6	0<V<=0.5
8			0.5<V<=0.75
9			V>=0.75
10		0.6<=S<0.8	0<V<=0.5
11			0.5<V<=0.75
12			V>=0.75
13		S>=0.8	0<V<=0.5
14			0.5<V<=0.75
15			V>=0.75
...	...	...	...
270	330<H<=350	S>=0.8	V>=0.75

After partitioning into *m* number of areas HSV information spaced image is further divided into *n* number of segments depending upon the number of pixels in each area. The areas which have more pixels are partitioned into more number of segments and the areas which have fewer pixels are divided into less number of segments. Through this procedure we are able to get the distribution characteristics for the color tone distributed in the corresponding image. After extracting this color distribution information from the image we calculated the maximum color occurrence in each segment by using hue histogram of the image and HSV value of that pixel is used to make a feature vector. Such operations are carried out for the entire image and then feature vector is computed.

Following equation is used to partition each area in n number of segments n can have different values for different areas.

$$n_i = (X_i / T) * TS \qquad 0<i<=m \qquad (6.1)$$

Where n_i represents the number of segments in area i, X_i represents the number of pixels in area i (where i ranges from 1 to m), T represents total number of pixels of the image and TS represents total number of required segments of the entire HSV image.

After the above step hue histogram is computed for each segment. From that histogram we can easily detect the each color occurrence in that segment. We have selected the maximum color occurrence from each segment and used its HSV value to compute the feature vector.

This feature vector is used for comparison with the images in the database and matching for each image is calculated using Minkowski Metric (LM norm) equation. Segments in each area of the query image are compared with the corresponding area of the database image and for this comparison we used Minkowski Metric as shown below:

$$d_M(i,j) = \left(\sum_{k=1}^{p} \left|(x_i^k - x_j^k)\right|^M\right)^{1/M} \quad (6.2)$$

Where, p is the dimension of the vector x_i^k is the kth element of x_i. Three special cases of the LM metric are of particular interest, namely, L =1, 2, 8.

2. Experiment and Results of Extracting Color Using Adaptive Segmentation for Image Retrieval

When matching calculation is completed, results are sorted in ascending order and the image which resembles more with the query image is brought to the top.

In order to test the algorithm proposed, a simple retrieval system was implemented on the web. Database taken from the website of Efficient Image Retrieval group, University of Washington is used for experimentation. It contains 11 sets of images taken from different parts of the world. Since the algorithm is based on the color information of the image so the rich color based images for experimentation were used.

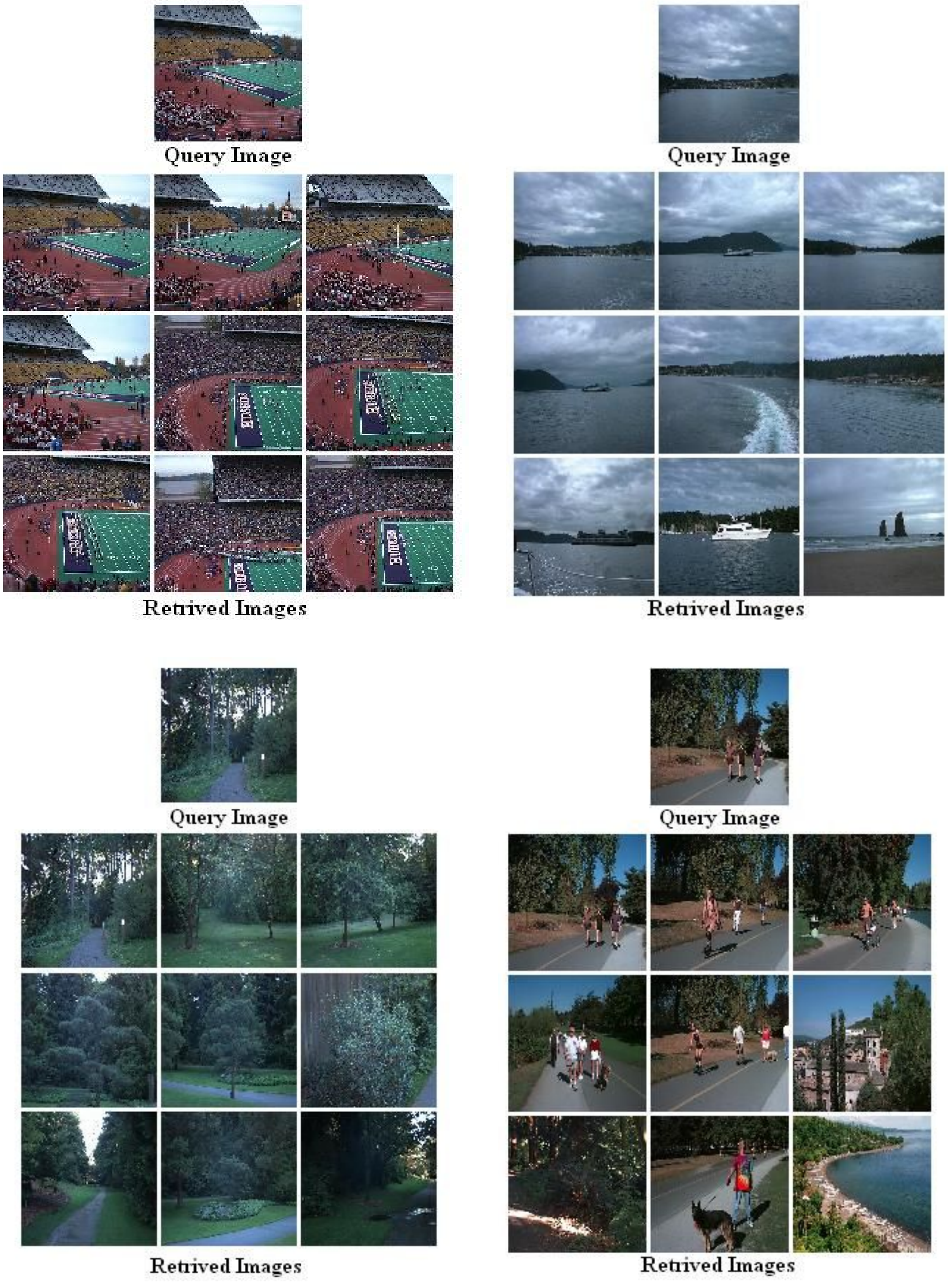

Figure 4.20: Experiment Results

V. Performance Analysis

A. Precision and Recall

In an Information Retrieval scenario, Precision is defined as the number of relevant documents retrieved by a search divided by the total number of documents retrieved by that search, and Recall is defined as the number of relevant documents retrieved by a search divided by the total number of existing relevant documents (which should have been retrieved).

In a classification task, a Precision score of 1.0 for a class C means that every item labeled as belonging to class C does indeed belong to class C (but says nothing about the number of items from class C that were not labeled correctly) whereas a Recall of 1.0 means that every item from class C was labeled as belonging to class C (but says nothing about how many other items were incorrectly also labeled as belonging to class C).

Often, there is an inverse relationship between Precision and Recall, where it is possible to increase one at the cost of reducing the other. For example, an information retrieval system (such as a search engine) can often increase its Recall by retrieving more documents, at the cost of increasing number of irrelevant documents retrieved (decreasing Precision). Similarly, a classification system for deciding whether or not, say, a fruit is an orange, can achieve high Precision by only classifying fruits with the exact right shape and color as oranges, but at the cost of low Recall due to the number of false negatives from oranges that did not quite match the specification.

Usually, Precision and Recall scores are not discussed in isolation. Instead, either values for one measure are compared for a fixed level at the other measure (e.g. precision at a recall level of 0.75) or both are combined into a single measure, such as the F-measure, which is the weighted harmonic mean of precision and recall, or the Matthews Correlation Coefficient.

In Information Retrieval contexts, Precision and Recall are defined in terms of a set of retrieved documents (e.g. the list of documents produced by a web search engine for a query) and a set of relevant documents (e.g. the list of all documents on the internet that are relevant for a certain topic).

Precision = ({relevant documents} ∩ {documents retrieved})/(documents retrieved)

Recall = ({relevant documents} ∩ {documents retrieved})/(relevant documents)

B. Performance Measures

Precision and recal measures are used to test the robustness and performance of proposed techniques. Proposed approaches have also been compared with already known algorithm [29]. Following table shows the measure of recall and precision of proposed algorithms and compare them with already known algorithm [29].

Table 4.6: Shows comparison of the proposed system with already known system.

	Recall	Precision
Known Algorithm [29]	0.5	0.24
Algorithm of Section E	0.5	0.26
Algorithm of Section F	0.5	0.27
Algorithm of Section G	0.5	0.30
Algorithm of Section H	0.5	0.28

By putting the recall constant at 0.5 for both the algorithm we checked the accuracy or precision of retrieval. We can clearly see that precision of the proposed algorithm is better than already known system. Performance comparison experiments are performed by using Professor wang's [24, 25] image database.

VI. Conclusion

In this dissertation, different methods to perform adaptive image segmentation are presented. These segmentation approaches are structured in three main stages:

- Convert image from RGB to HSV, as explained in section D of chapter III.
- Divide the image into different areas and get the pixel information about each area.
- Extract segments adaptively from each and every area based on the pixel information of that area.
- The areas which have more pixels are divided into more number of segments and the areas which have less pixels are divided into less number of segments.
- Feature vectors are constructed by using the above information.
- Feature vector of query image is compared with each feature vector of the database images.
- Comparison of feature vectors are performed by using two different similarity measures i.e. Euclidean distance and Minkowski metric.
- Results are obtained after feature comparison.
- Last but not the least, all the related images are sorted in ascending order and most relevant images are brought to the top and displayed on the screen.

The First stage consists of using the standard method for converting RGB color space to HSV color space using the equations as explained in section D of chapter III. Second stage divides the image into different areas by using different ranges of Hue, Saturation and/or Value to get the color information of the image.

Finally the segmentation is performed adaptively by calculating the information about the number of pixels in each area. This is our main idea behind each and every algorithm presented in this dissertation.

The first algorithm performs adaptive segmentation on the HSV cone by using only Hue and Value information, the second algorithm uses Hue and Saturation, third algorithm uses all the three components of the HSV cone to segment the image. Finally we have used a Hue histogram after segmenting HSV cone using hue, saturation and value components.

For testing algorithms robustness and efficiency we developed small applications for image retrieval. Different set of image databases are used containing different images from various categories. The experiment results show that proposed systems retrieval results efficiently and robustly.

Finally the efficiency of the proposed algorithms is tested by using precision and recall measures. Precision and recall measures of our proposed methods show superior performance than the already known HSI Indexing algorithm, which proves that the proposed algorithms are more efficient and robust in retrieving the results precisely.

References

1. Database Techniques for Pictorial Applications, Lecture Notes in Computer Science, A. Blaser, ed., vol. 81, Springer Verlag GmbH, 1979.
2. Y. Rui, T.S. Huang, and S.-F. Chang, "Image Retrieval: Current Techniques, Promising Directions, and Open Issues", J. Visual Comm. and Image Representation, Vol. 10, pp. 39-62, 1999.
3. P. Aigrain, H. Zhang, and D. Petkovic, "Content-Based Representation and Retrieval of Visual Media: A State of the Art Review", Multimedia Tools and Applications, Vol. 3, pp. 179-202, 1996.
4. R. Brunelli, O. Mich, and C.M. Modena, "A Survey on the Automatic Indexing of Video Data", J. Visual Comm. and Image Representation, Vol. 10, pp. 78-112, 1999.
5. S. Ornager, "Image Retrieval: Theoretical and Empirical User Studies on Accessing Information in Images", Proc. 60th Am. Soc. Information Science Ann. Meeting, Vol. 34, pp. 202-211, 1997.
6. L. Armitage and P. Enser, "Analysis of User Need in Image Archives", J. Information Science, Vol. 23, pp. 287-299, 1997.
7. Robert M. Haralick and Linda G. Shapiro. Image segmentation techniques. Computer Vision, Graphics and Image Processing, 1985.
8. Nikhil R. Pal and Sankar K. Pal. "A review on image segmentation techniques", Pattern Recognition, 1993.
9. Berthold K. P. Horn. Robot Vision. MIT Press, Cambridge, 1986.
10. J. Ross Beveridge, Joey Griffith, Ralf R. Kohler, Allen R Hanson, and Edward M Riseman. Segmenting images using localized histograms and region merging. International Journal of Computer Vision, 1989.
11. M. J. Swain and D. H. Ballard, "Color Indexing", International Journal of Computer Vision Vol. 7, pp. 11-32, 1991.
12. B. V. Funtand G.D. Finlayson, "Color constant color indexing", IEEE Trans. on

Pattern Analysis and Machine Intelligence, Vol. 17, pp. 522-529, 1995.

13. B. M. Mehtre, M. S. Kankanhalli, A. D. Narsimhalu and G. C. Man, "Color matching for image retrieval", Pattern Recognition Letter, Vol. 16, pp. 325-331, 1995.
14. M. Safar, C. Shahabi and X. Sun, "Image retrieval by shape: a comparative study", ICME 2000, Vol. 1, pp. 141-144, 2000.
15. A.K. Jain, "Fundamental of Digital Image Processing", Prentice Hall International, 1989.
16. A. Khotanzad and Y.H. Hong, "Invariants Image Recognition by Zernike Moments", IEEE Trans. on Pattern Analysis and Machine Intelligence, Vol. 12, pp. 489-497, 1990.
17. S. Belongie, C. Carson, H. Greenspan and J. Malik, "Color and Texture-based Image Segmentation Using EM and Its Application to Content-based Image Retrieval", Computer Science Division, University of California at Berkeley, ICCV'98, 1998
18. R. Chellappa and S. Chatterjee, "Classification of Textures Using Gaussian Markov Random Fields", IEEE Trans. on Acoustics, Speech, and Signal Processing, Vol. ASSP-33, pp. 959, 1985
19. B. S. Manjunath and W. Y. Ma, "Texture Features for Browsing and Retrieval of Image Data", IEEE Trans. on Pattern Analysis and Machine Intelligence, Vol.18, pp.837-842, 1996
20. J. Huang, "Color-Spactial Image Indexing and Application", Ph.D. thesis in Cornell Univ., 1998.
21. G. Pass and R. Zabih, "Comparing images using joint histogram", Multimedia Systems, Vol.7, pp. 234-240, 1999
22. David A. Forsyth, Jean Ponce, "Computer Vision: A Modern Approach", Prentice Hall; US Ed edition, pp. 53-91, 2002.
23. Bo Di, "An efficient image retrieval approach base on color clustering", Third International Conference on Intelligent Information Hiding and Multimedia Signal Processing, 2007. IIHMSP 2007.

24. Jia Li, James Z. Wang, "Automatic linguistic indexing of pictures by a statistical modeling approach," IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 25, pp. 1075-1088, 2003.
25. James Z. Wang, Jia Li, Gio Wiederhold, "SIMPLIcity: Semantics-sensitive Integrated Matching for Picture LIbraries," IEEE Trans. on Pattern Analysis and Machine Intelligence, Vol 23, pp. 947-963, 2001.
26. Muhammad Riaz, Gwangwon Kang, Younghee Kim, Sungbum Pan, and Jongan Park, "Efficient Image Retrieval Using Adaptive Segmentation of HSV Color Space", International Conference on Computational Sciences and its Applications, 2008. ICCSA'08, June 30-July 3 2008.
27. J. Chen, T.N. Pappas, A. Mojsilovic, and B.E. Rogowitz, "Adaptive image segmentation based on color and texture", Proc. ICIP 2002, Rochester, NewYork, Sept. 2002.
28. Muhammad Riaz, An Youngeun, and Jongan Park, "Feature Vector Extraction System Based on Adaptive Segmentation of HSV Information Space", 11th International Conference on Computer Modeling and Simulation, 2009. UKSIM2009, March 25-27 2009.
29. Jongan Park, Sungkwan Kang, Ilhoe Jeong, Waqas Rasheed, Seungjin Park, and Youngeun An, "Web based Image Retrieval System Using HSI Color Indexes", International Conference on Intelligent Computin, ICIC 2007, August 21-24 2007.
30. Muhammad Riaz, Kim Pankoo, and Jongan Park, "Extracting Color Using Adaptive Segmentation For Image Retrieval", 2009 International Symposium on Applied Computing and Computational Sciences, ACCS 2009, April 24-26 2009.
31. Proc. US NSF Workshop Visual Information Management Systems, R. Jain, Ed. 1992.

(별 지)

저작물 이용 허락서

학 과	정보통신공학과	학 번	20077789	과 정	석사
성 명	영문 : Muhammad Riaz				
주 소	광주광역시 동구 서석동 375 조선대학교 전자정보공과대학 821호				
연락처	E-MAIL : gr8riaz@gmail.com				
논문제목	영문 : Adaptive Multi-dimension Segmentation of HSV Cone for Image Retrieval				

본인이 저작한 위의 저작물에 대하여 다음과 같은 조건아래 조선대학교가 저작물을 이용할 수 있도록 허락하고 동의합니다.

- 다 음 -

1. 저작물의 DB구축 및 인터넷을 포함한 정보통신망에의 공개를 위한 저작물의 복제, 기억장치에의 저장, 전송 등을 허락함
2. 위의 목적을 위하여 필요한 범위 내에서의 편집형식상의 변경을 허락함. 다만, 저작물의 내용변경은 금지함.
3. 배포전송된 저작물의 영리적 목적을 위한 복제, 저장, 전송 등은 금지함.
4. 저작물에 대한 이용기간은 5년으로 하고, 기간종료 3개월 이내에 별도의 의사표시가 없을 경우에는 저작물의 이용기간을 계속 연장함.
5. 해당 저작물의 저작권을 타인에게 양도하거나 또는 출판을 허락을 하였을 경우에는 1개월 이내에 대학에 이를 통보함.
6. 조선대학교는 저작물의 이용허락 이후 해당 저작물로 인하여 발생하는 타인에 의한 권리 침해에 대하여 일체의 법적 책임을 지지 않음
7. 소속대학의 협정기관에 저작물의 제공 및 인터넷 등 정보통신망을 이용한 저작물의 전송출력을 허락함.

2009 년 6 월 1일

저작자: Muhammad Riaz (서명 또는 인)

조선대학교 총장 귀하